

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

CLAIMS

1. (Currently Amended) In a **Java JAVA** computing environment, a method of generating optional attributes in a **Java JAVA** class file, said method comprising:

receiving as input a **Java JAVA** runtime **environment** optimization **information**;

generating one or more optional attributes based on said Java runtime **environment** optimization **information**; and

writing said one or more optional attributes in an attribute table portion of a Java class file.
2. (Cancelled)
3. (Original) A method as recited claim 1, wherein said method further comprises:

generating computer program code that implements an application programming interface suitable for loading said one or more optional attributes.
4. (Original) A method as recited claim 3, wherein said application programming interface can be used to read said one or more optional attributes from said Java class file.
5. (Currently Amended) A method as recited claim 4, wherein said application programming interface includes functions that can be used to read first, last, and next optional attributes in said **Java JAVA** class file.
6. (Currently Amended) A method as recited claim 4, wherein said application programming interface includes a function suitable for finding an optional attribute in said **Java JAVA** class file.

7. (Currently Amended) A method as recited claim 1, wherein said **Java JAVA** runtime environment optimization information is stored in a database.
8. (Currently Amended) A method as recited in claim 7, wherein said database is generated by a compiler extension or a software tool suitable for analyzing a **Java JAVA** application.
9. (Original) A method as recited in claim 7, wherein said database is stored in a runtime performance manager that can interact with software modules that generate and load said one or more optional attributes.
10. (Original) A method as recited in claim 7, wherein said method further comprises:
updating said database to reflect generation of said one or more optional attributes.
11. (Currently Amended) In a **Java JAVA** computing environment, a **Java JAVA** optional attribute generator **computer-implemented method** suitable for generation of optional attributes in a **Java JAVA** class file, said **Java JAVA** optional attribute generator **computer-implemented method** operating to:
receive as input a **Java JAVA** runtime **environment** optimization **information**;
generate one or more optional attributes based on said **Java JAVA** runtime **environment** optimization **information**; and
write said one or more optional attributes in an attribute table portion of a **Java JAVA** class file.
12. (Currently Amended) A **Java JAVA** optional attribute generator as recited in claim 11, wherein said **Java JAVA** optional attribute generator **computer-implemented method** operates to generate computer program code that implements an application programming interface suitable for loading said one or more optional attributes.

13. (Currently Amended) A **Java JAVA** optional attribute generator **computer-implemented method** as recited in claim 11, wherein an application programming interface can be used to read said one or more optional attributes from said **Java JAVA** class file.

14. (Currently Amended) A **Java JAVA** optional attribute generator **computer-implemented method** as recited in claim 11, wherein said **Java JAVA** runtime **environment** optimization **information** is stored in a database.

15. (Currently Amended) A **Java JAVA** optional attribute generator **computer-implemented method** as recited in claim 11, wherein said database is generated by a compiler extension or a software tool suitable for analyzing a **Java JAVA** application.

16. (Currently Amended) A **Java JAVA** optional attribute generator **computer-implemented method** as recited in claim 11,

wherein said database is stored in a runtime performance manager that can interact with software modules that generate and load said one or more optional attributes.

17. (Currently Amended) A **Java JAVA** optional attribute generator as recited in claim 11, wherein said optional attribute generator **computer-implemented method** operates to update said database to reflect generation of said one or more optional attributes.

18. (Currently Amended) A **Java JAVA** optional attribute generator as recited in claim 11, wherein said optional attribute generator **computer-implemented method** operates to generate a description of an optional attribute.

19. (Currently Amended) A **Java JAVA** optional attribute generator **computer-implemented method** as recited in claim 18, wherein said description is in XML format.

20. (Currently Amended) A computer readable medium including computer program code for generating optional attributes in a **Java JAVA** class file, said computer readable medium comprising:

computer program code for receiving as input ~~a-**Java JAVA**~~ runtime **environment** optimization **information**;

computer program code for generating one or more optional attributes based on said Java runtime **environment** optimization **information**; and

computer program code for writing said one or more optional attributes in an attribute table portion of a Java class file.

21. (Original) A computer readable medium as recited in claim 20, wherein said method further comprises:

generating computer program code that implements an application programming interface suitable for loading said one or more optional attributes.

22. (Currently Amended) A computer readable medium as recited in claim 21, wherein said **Java JAVA** runtime **environment** optimization **information** is stored in a database.

23. (Currently Amended) A computer readable medium as recited in claim 22, wherein said database is generated by a compiler extension or a software tool suitable for analyzing a **Java JAVA** application.

24. (Original) A computer readable medium as recited in claim 22, wherein said database is stored in a runtime performance manager that can interact with software modules that generate and load said one or more optional attributes.

25. (Original) A computer readable medium as recited in claim 24, wherein said method further comprises:

updating said database to reflect generation of said one or more optional attributes.